

# REPORTING

## MASTER INTEGRATION DE SYSTEMES LOGICIELS



Présenté par : Seddik OUISS  
Tuteur : Julien Aymard  
Encadrant : Jean-Luc Massat

SEDDIK OUISS  
SMART TRADE TECHNOLOGY

2014/2015

# CLAUSE DE CONFIDENTIALITÉ

Les informations contenues dans ce document sont confidentielles et ne doivent en aucun cas être copiées ou diffusées sans l'autorisation de Smart Trade Technologies.

## Remerciement

Tout d'abord, je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé à la rédaction de ce rapport.

Je tiens à remercier vivement mon tuteur de stage, Mr Julien Aymard, responsable au sein de Smart Trade Technologies du projet R5, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce aussi à sa confiance, j'ai pu m'accomplir totalement dans mes missions avec son aide précieuse dans les moments les plus délicats.

Je remercie également toute l'équipe R5, pour le temps passé à m'expliquer le travail déjà réalisé dans le projet R5, me facilitant ainsi l'intégration dans leur équipe.

Je remercie également Mr Éric Deshayes, responsable à Smart Trade Technologies, pour ses directives techniques ainsi que ses conseils sur l'intégration de mon sujet dans la plateforme technique de Smart Trade.

J'adresse mes remerciements à Mr Jean-Luc Massat, mon encadrant, ainsi que tous les professeurs d'Aix-Marseille Université qui m'ont enseigné et partagés leurs connaissances durant mon cursus.

Je remercie bien sûr M. David VINCENT de m'avoir accepté dans son entreprise, ainsi que l'ensemble des employés de Smart Trade Technologies pour l'aide, le soutien et l'ambiance de travail chaleureuse qu'ils m'ont apportés.

Enfin, je tiens à remercier toutes les autres personnes qui m'ont conseillé et aidé et soutenu durant toute la durée de ce stage: ma famille, mes amis et mes camarades proches.

## Fiche technique

<b>Etudiant:</b>	Seddik Ouiss
<b>Année:</b>	2015
<b>Raison sociale de l'entreprise:</b>	Smart Trade Technologies
<b>Tuteur de stage:</b>	Julien Aymard
<b>Enseignant responsable:</b>	Jean-Luc Massat
<b>Sujet du stage:</b>	L'objectif du stage est de mettre en place une infrastructure facilitant le développement d'une application de <i>reporting</i> .
<b>Plate-forme informatique:</b>	Environnement Smart Trade (conteneur léger basé sur Spring)
<b>Outils &amp; logiciels:</b>	Eclipse, Java, Spring, PonySDK (Basé sur GWT), Pivot4J, Olap4J, Mondrian, Hibernate, Pentaho Kettle.

# SOMMAIRE

Remerciement .....	2
Fiche technique .....	3
Préface .....	5
Présentation .....	6
Smart Trade Technologies.....	6
Aix-Marseille Université.....	10
Travail réalisé.....	11
Introduction .....	11
Contexte .....	11
Problématique.....	11
Etude préalable .....	12
Besoins fonctionnels .....	15
Contraintes techniques .....	15
Solution proposée .....	16
Infrastructure proposée.....	17
Avantages de la solution proposée.....	19
Inconvénients de la solution proposée .....	19
Modèle en constellation .....	20
Front end .....	21
Logiciels, outils informatiques, protocoles et standards utilisés .....	25
Bilan du travail.....	31
Problèmes rencontrés .....	31
Bilan du travail en entreprise.....	32
Bilan du travail personnel .....	33
Conclusion .....	34
Lexique .....	35
Annexes .....	36
MOLAP, ROLAP et HOLAP - Bases de données multidimensionnelles .....	36
Modèle en flocon .....	38
Mondrian: Déclaration et mapping d'un cube.....	39
Table des figures .....	40
Webographie .....	41

## Préface

Dans le cadre de l'obtention du Master 2 professionnel en Intégration de systèmes logiciels (ISL), la formation permet à chaque étudiant d'effectuer un stage de fin d'études idéalement dans le domaine qui l'intéresse pour une durée de 5 à 6 mois.

Désirant mettre en pratique les connaissances acquises durant ma formation, j'ai ressenti le besoin de m'orienter vers une entreprise dont la rigueur et la qualité sont les principales priorités. J'ai commencé mes recherches auprès des entreprises du secteur dès octobre 2014. Elles ont abouti au mois de mars lorsque l'entreprise Smart Trade Technologies a accepté de m'accueillir en tant que stagiaire d'avril à septembre 2015.

L'objectif de mon stage était d'ajouter à Smart Trade la possibilité de faire de la « Business Intelligence » ou plus précisément du « reporting » non statique, c'est-à-dire permettre aux clients Smart Trade d'exploiter l'énorme masse de données stocker par les outils Smart Trade et de pouvoir l'utiliser à des fins commerciales ou stratégiques.

Pour ce faire, j'ai dû étudier ce qui se faisait tout d'abord chez Smart Trade et ensuite ce qui se faisait de mieux dans ce domaine. Ceci m'a amené à une phase d'étude préalable dans laquelle j'ai réalisé plusieurs prototypes techniques qui ont permis de délimiter le sujet de stage.

Dans ce rapport, après la présentation de l'entreprise et des objectifs principaux de mon stage, j'aborderai l'essentiel du travail réalisé.

# Présentation

## Smart Trade Technologies

Smart Trade Technologies fut créée en 1999, par Harry GOZLAN qui occupe aujourd'hui le poste de président exécutif au sein de l'entreprise, et David VINCENT au poste de CEO 2. Smart Trade fournit des systèmes de gestion de la liquidité sophistiqués, permettant aux banques de développer des plateformes de transaction multi-actifs. Ces systèmes regroupent des liquidités de dizaines de sources pour créer un carnet d'ordres uniques, de distribuer des prix personnalisés aux clients. De recevoir et de gérer les flux d'ordres de clients, internaliser des liquidités aux sites externes. Ils gèrent l'état du cycle de vie de l'ordre, en mettant en œuvre le routage intelligent des ordres et un des algorithmes propriétaires, en reliant électroniquement à un certain nombre de lieux, indépendamment des protocoles de messagerie. Smart Trade est implantée dans plusieurs villes à travers le monde afin d'être plus près de ses clients, à savoir Londres, New York, Tokyo, et Aix-en-Provence où se déroule mon stage comportant plus d'une quarantaine de personnes travaillant sur différents projets.

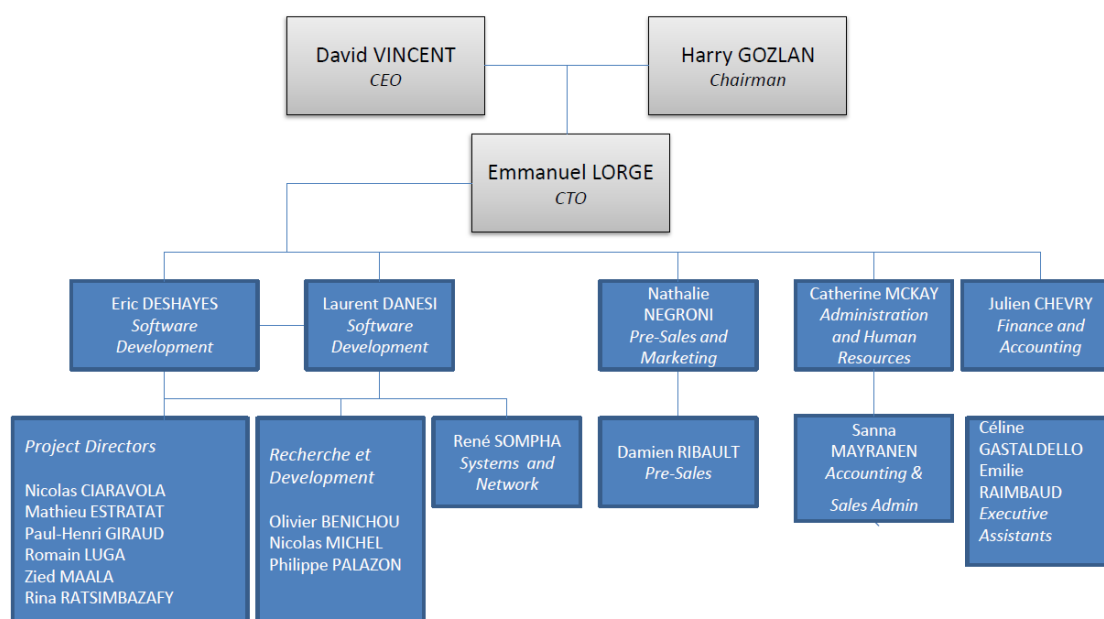


Figure 1: Organigramme Smart Trade

Smart Trade propose aussi un LMS (Liquidity Management System) sophistiqué, permettant aux banques:

- La gestion de leurs liquidités.
- L'optimisation des processus de trading.
- La gestion des différentes règles de management de prix entre les ventes.
- Faire gagner du temps aux traders.
- Faire des ordres et des trades sur tout type d'instruments (matières première, monnaie, actions, etc.).



## Organisation des équipes à Smart Trade

A Smart Trade il y a plusieurs équipes comme le montre le schéma ci-dessous.

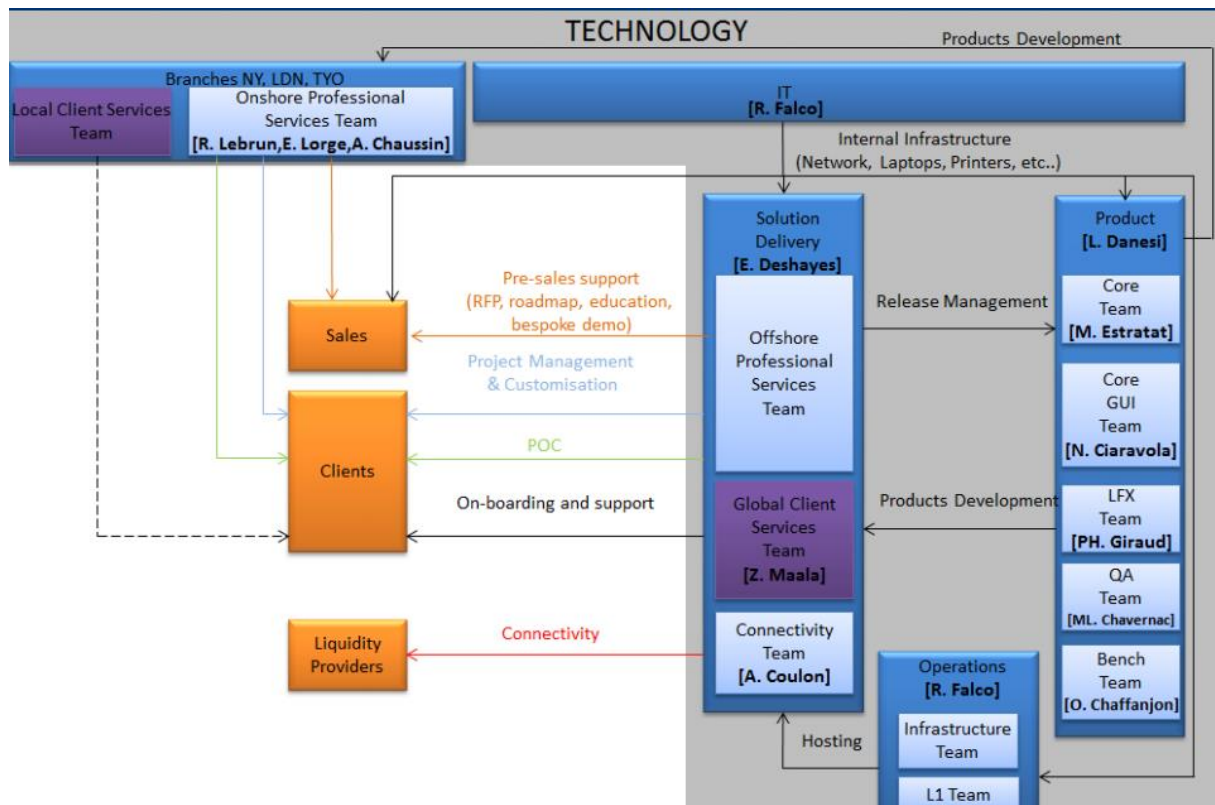


Figure 2: Liste des projets Smart Trade

- CORE UI: fournit tous les éléments nécessaires pour le côté front end, c'est-à-dire gère l'ensemble des bibliothèques des tous les composants graphiques de Smart Trade.
- CORE: fournit tous les éléments nécessaires pour le côté back end, c'est-à-dire gère l'ensemble des bibliothèques de tous les composants métiers de Smart Trade.
- LFX: LFX 3 étant un produit Smart Trade, cette équipe s'occupe essentiellement de l'amélioration du produit.
- QA: contrôle la qualité des logiciels de Smart Trade du point de vue fonctionnel.
- Bench: contrôle la capacité et les ressources utilisées par les logiciels afin de les optimiser.

L'équipe, R5, où je suis affecté et qui s'occupe d'un projet client du même nom ne figure pas dans ce schéma. En effet, le projet R5 est récent et repose sur les fondations de Smart Trade qui sont CORE et CORE UI.

## Outils et services

Pour le bon déroulement du travail et pour produire un résultat de qualité l'entreprise nous mis à disposition plusieurs outils permettant de simplifier le travail qui sont:

- TimeSheet: comme outil de gestion des ressources, développé en interne.
- Bamboo: pour les tests d'intégration continue.
- CVS et GitLab: pour la gestion de versions des projets Smart Trade.
- Confluence: la base de connaissances de l'entreprise.
- Artifactory: la base de données des librairies internes et externes utilisées dans l'entreprise.

Et pour faciliter la communication au sein de l'entreprise, et avec les clients, les outils ci-dessous sont aussi utilisés:

- JIRA: système de suivi de bug, de gestion des incidents et de gestion de projets.
- Confluence: la base de connaissances de l'entreprise.
- Hangout: messagerie instantanée.
- mais aussi les mails, fax, et téléphone.

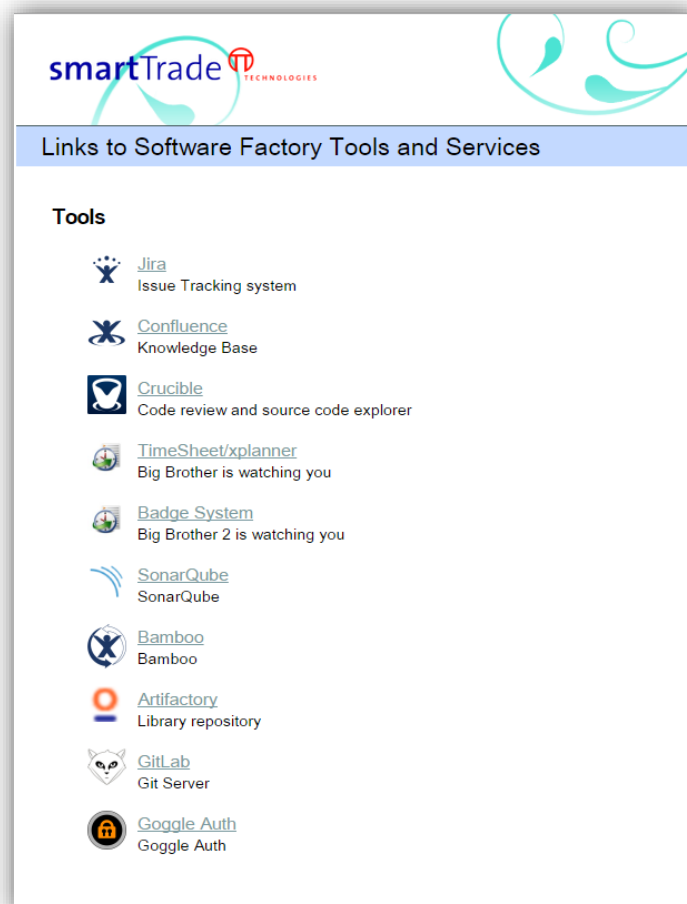


Figure 3: Outils et service interne de Smart Trade

## Aix-Marseille Université

Le pôle Aix-Marseille Université a été créé le 24 août 2011. Il regroupe l'université de Provence, de la Méditerranée et Paul-Cézanne. C'est aujourd'hui une des plus jeunes universités de France, c'est aussi la plus grande de par le nombre de ses étudiants, de son personnel et de son budget. C'est également la plus grande université francophone. Autant d'atouts, au-delà des résultats d'ores et déjà notables dans le domaine de l'enseignement et de la recherche, qui font d'Aix-Marseille Université un établissement d'enseignement supérieur et de recherche d'excellence.

Il possède 7 départements:

- Mathématiques
- Physique
- Chimie
- Mécanique
- Biologie
- Sciences, arts et techniques de l'image et du son.
- Informatique.

Le département informatique possède plusieurs spécialités au niveau du master, dont le FSIL « Fiabilité, Sécurité et Intégration Logicielle » d'où je viens. La spécialité FSIL a pour but de former des étudiants afin qu'ils puissent concevoir, réaliser, valider et mettre en exploitation des logiciels. Pour cela, les concepts, les outils et les méthodes liés au développement de logiciels sont mis en valeur. Les aspects de qualité, de sécurité et de bonne intégration dans des environnements informatiques complexes constituent également les points forts de la formation.

# Travail réalisé

## Introduction

Lorsque j'ai postulé à Smart Trade pour effectuer un stage, le sujet n'était pas déterminé. On m'a proposé un sujet de stage sur le reporting. J'ai accepté ce sujet, car il me permet de découvrir un domaine très utilisé dans la plupart des projets informatiques. L'objectif du stage est donc de mettre en place une infrastructure facilitant le développement d'une application de reporting ainsi qu'une application web de reporting ergonomique.

## Contexte

Ce projet est un projet de recherche et développement qui a pour objectif, sur le long terme, d'être intégré aux différents projets Smart Trade. Soit en les remplaçant soit en les complétant.

## Problématique

Les différents outils de « reporting » existants à Smart Trade ne sont pas très flexibles. En effet, les rapports générés sont statiques et nécessitent une modification dans le code dans le cas où par exemple on voudrait les modifier ou en ajouter des nouveaux.

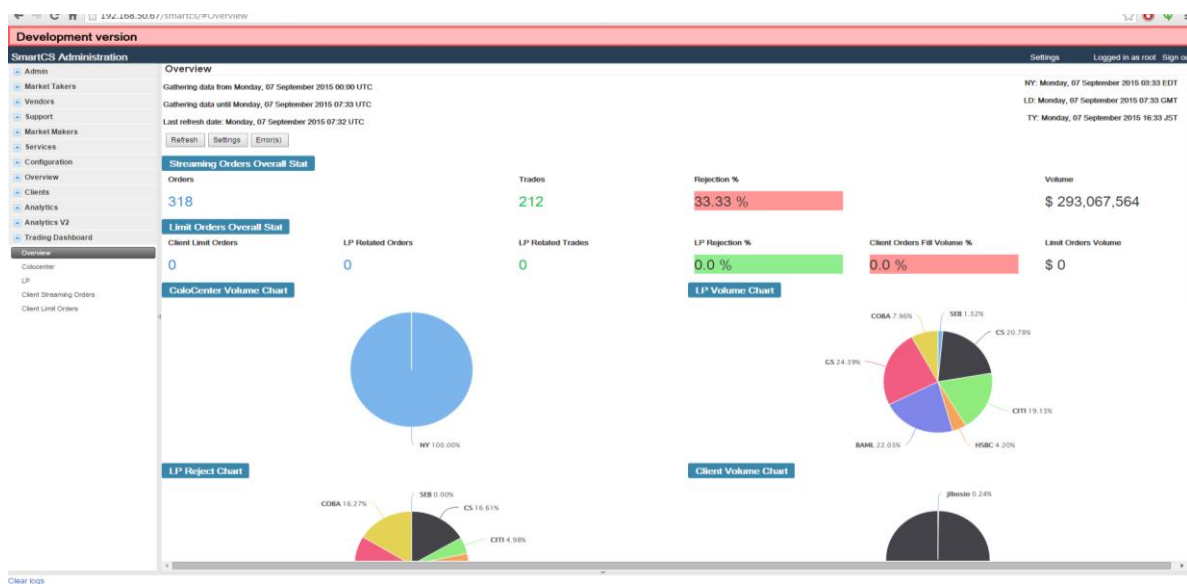


Figure 4: Exemple d'outils de reporting existant

## Etude préalable

Tout d'abord, j'ai fait plusieurs recherches sur les différents outils de « reporting ». J'ai testé notamment les outils de reporting comme BIRT ou encore JasperSoft. Ces derniers ne sont pas adaptés aux besoins de Smart Trade.

Après quelques réunions, il a été décidé de mettre en place un entrepôt de données, j'ai alors fait une étude sur la manière dont ces derniers sont conçus. Après cette étude, j'ai constaté que différentes approches sont possibles (voir annexe: MOLAP, ROLAP et HOLAP - Bases de données multidimensionnelles). J'ai choisi l'approche ROLAP (voir annexe: ROLAP: Relational Online Analytical Processing). En effet, c'est la solution la plus évoluée dans le monde open source. Avec Mondrian comme Moteur OLAP qui, par ailleurs, est utilisé par la plupart des outils open source de la Business Intelligence comme Pentaho BI Server ou encore JasperReports Server. De plus, Mondrian peut être utilisé comme fournisseur (provider) XMLA.

Ensuite, il fallait faire une étude sur les technologies faisant interface entre le moteur ROLAP et une application Java quelconque. La référence étant Olap4J qui est une API commune pour les données multidimensionnelles. Cette dernière permet de se connecter à n'importe quelle source de données disposant d'un driver JDBC. Elle fournit également un pilote permettant de se connecter à n'importe quel fournisseur respectant le standard XMLA.

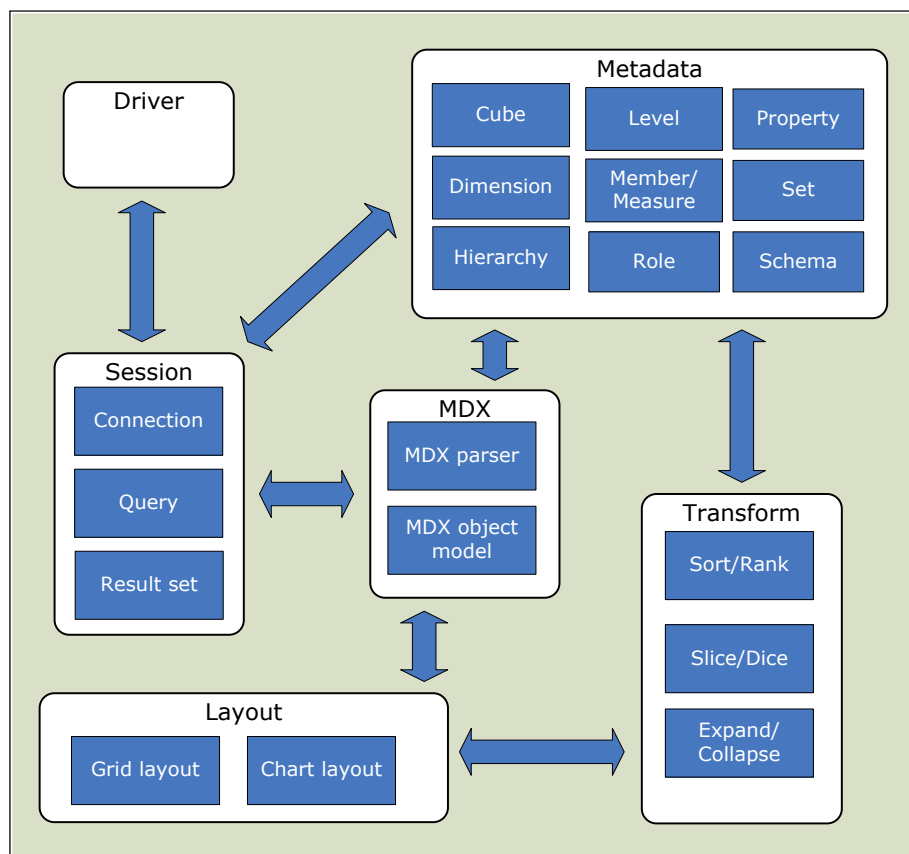


Figure 5: Structure de l'API Olap4J

Les « Metadata » forment le modèle de l'API Olap4J. Ils sont structurés de manière arborescente. La description de cette structure est adaptée au format XML. En effet, le moteur Mondrian, par exemple, utilise un fichier XML pour la description de la structure multidimensionnelle qu'il publie.

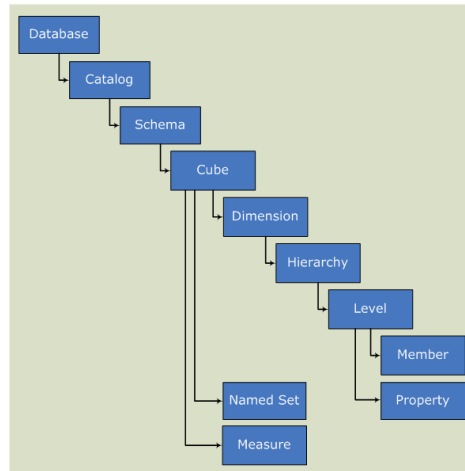


Figure 6: Metadata Olap4J

Les « Metadata » les plus importantes sont:

- Les cubes: Un cube représente un ensemble de mesures organisées selon un ensemble de dimensions.
- Les dimensions: une dimension est un axe d'analyse c'est-à-dire une base sur laquelle seront analysées les données. Une dimension possède des instances, également appelées membres. Chaque membre appartient à un niveau hiérarchique.
- Les mesures: une mesure est l'élément de donnée que l'on analyse.

J'ai également utilisé l'API Pivot4J qui permet de construire une application d'analyse avec des tableaux croisés dynamiques (tableau pivot) peu importe la technologie Java utilisée (JSP, GWT, JSF, etc.). Pivot4J fonctionne au-dessus d'Olap4J et est un des remplaçants de JPivot. En effet, pendant longtemps, JPivot, a été la seule bibliothèque open-source permettant de faire des applications front end d'analyse. Le défaut de Pivot4J étant l'obligation d'utiliser des pages JSP (Java Server Page).

Buyer			Product		Measures			
(All)	Counter	user	(All)	Product	Quantity Traded	MIN Quantity Traded	Units Trade	Trade Volume
All Buyers	All Buyers		All Products		598,900,000	200,000	603	\$ 409,510.30
	Bank Of China	Kim DAVIS	All Products		8,000,000	1,000,000	8	\$ 281.28
			All Products	USD/CNY	2,000,000	1,000,000	2	\$ 12.40
				USD/PHP	6,000,000	1,000,000	6	\$ 268.88
	Bank Of China	Kim DAVIS	All Products		8,000,000	1,000,000	8	\$ 281.28
			All Products	USD/CNY	2,000,000	1,000,000	2	\$ 12.40
				USD/PHP	6,000,000	1,000,000	6	\$ 268.88
	Bank Of India		All Products		32,000,000	1,000,000	32	\$ 2,012.16
			All Products	USD/INR	32,000,000	1,000,000	32	\$ 2,012.16
	HSBC		All Products		44,000,000	1,000,000	44	\$ 206.19
			All Products	USD/BRL	43,000,000	1,000,000	43	\$ 143.19
				USD/INR	1,000,000	1,000,000	1	\$ 63.00
	JPMorgan		All Products		19,000,000	1,000,000	19	\$ 143.67
			All Products	USD/CNY	2,000,000	1,000,000	2	\$ 12.62
				USD/IDR	7,000,000	1,000,000	7	\$ 94.05
				USD/MYR	10,000,000	1,000,000	10	\$ 37.00
	R5Admin		All Products		296,300,000	600,000	297	\$ 291,502.64
			All Products	USD/BRL	6,600,000	600,000	7	\$ 21.12
				USD/CNY	1,000,000	1,000,000	1	\$ 6.02
				USD/IDR	10,000,000	1,000,000	10	\$ 130.90
				USD/INR	15,000,000	1,000,000	15	\$ 937.01
				USD/KRW	256,700,000	700,000	257	\$ 290,096.04
				USD/PHP	7,000,000	1,000,000	7	\$ 311.57
	Royal Bank of Scotland		All Products		199,600,000	200,000	203	\$ 115,364.35
			All Products	USD/KRW	97,800,000	800,000	98	\$ 110,663.64
				USD/PHP	101,800,000	200,000	105	\$ 4,700.71

Figure 7: Tableau croisé dynamique (avec Pivot4J et PonySDK)

## Besoins fonctionnels

Les principaux besoins fonctionnels sont:

- Générateur de rapport qui couvre le métier.
- Modification simple des rapports.
- application ergonomique

## Contraintes techniques

Les contraintes techniques imposées sont:

- Langage de programmation JAVA
- Conteneur léger Spring
- PonySDK en front end
- Interface graphique ergonomique
- Base de données MySQL
- API et Framework open source avec des licences non restrictives pour les entreprises.
- Respect des standards de la Business Intelligence



## Solution proposée

La solution proposée est:

- la mise en place d'un entrepôt de données avec MySQL.
- la mise en place d'une ETL avec Pentaho Kettle
- le déploiement d'un moteur OLAP avec Mondrian
- le développement d'un client OLAP sur SmartCC

Cette solution ressemble à celle proposée par Pentaho avec la solution « Pentaho BI Server ».

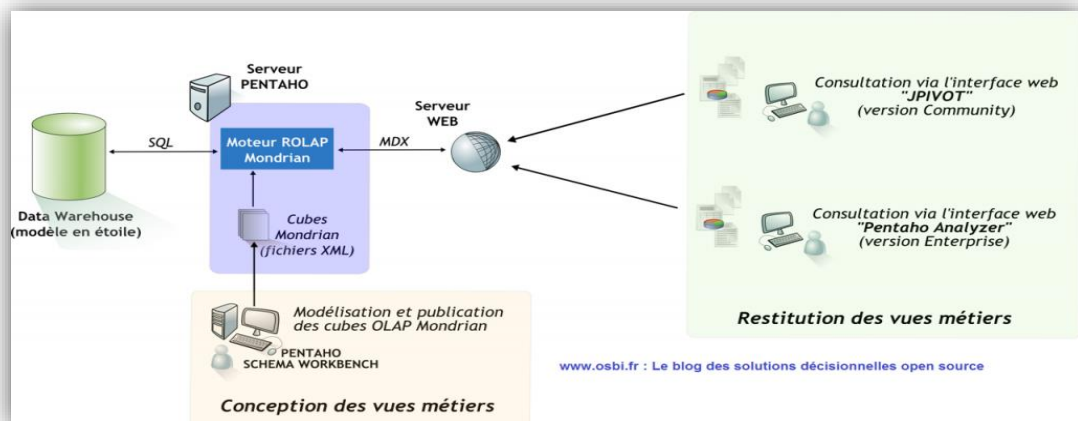


Figure 8: Infrastructure Pentaho BI Server

## Infrastructure proposée

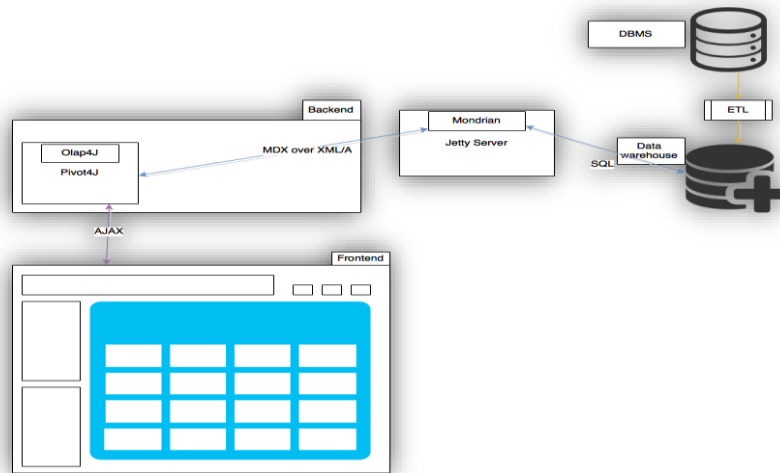


Figure 9: Infrastructure proposée

Kettle Pentaho sera mis en place en tant qu'ETL et va transformer de manière quotidienne les données présentes dans la base de données fonctionnelle pour ensuite les insérer dans l'entrepôt de données.

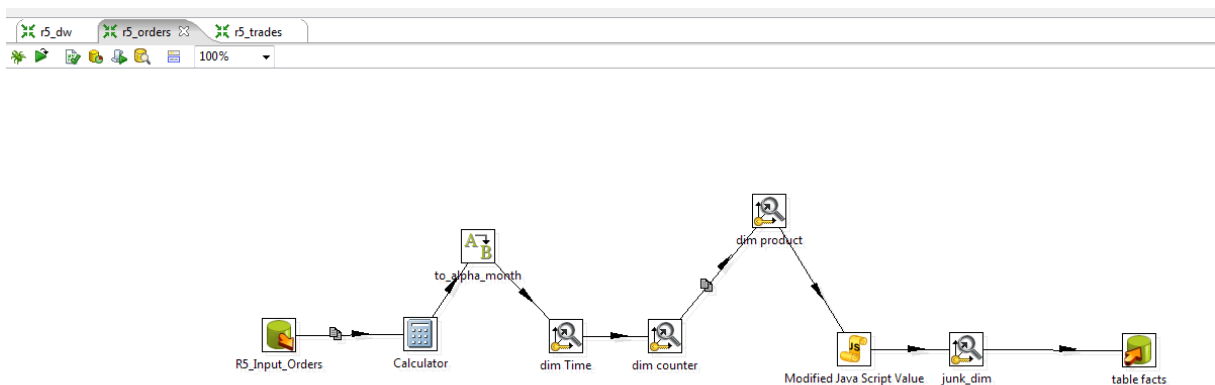


Figure 10: ETL (Insertion dans le cube des ordres)

Après la transformation des données, Mondrian pourra alors être interrogé, via XMLA, après que son cache soit vidé. L'utilisateur pourra, par exemple, consulter un rapport.

La communication entre le client et le serveur se fait avec AJAX.

## Diagramme de déploiement

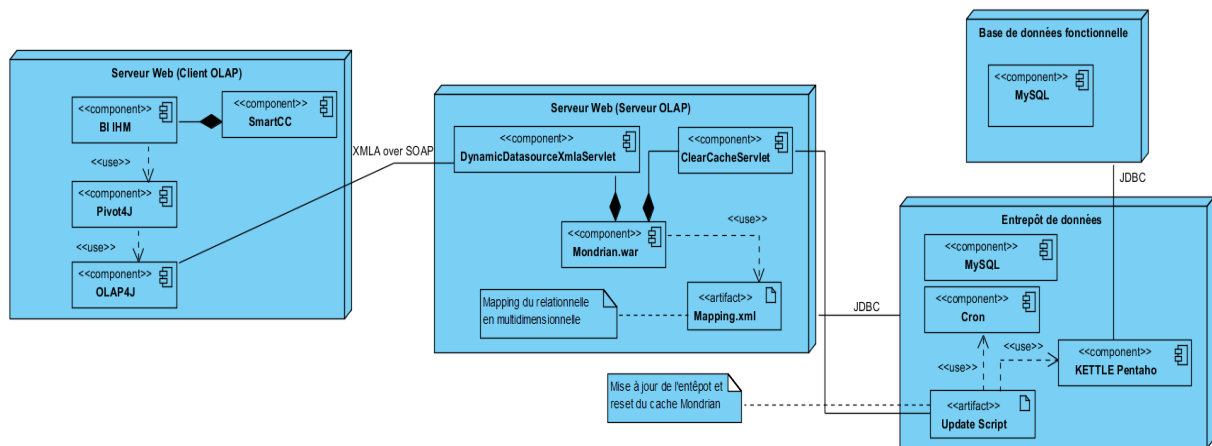


Figure 11: Diagramme de déploiement

## Avantages de la solution proposée

La solution proposée est très flexible. En effet, la connexion entre le client OLAP et le moteur OLAP (Mondrian) s'effectue au travers du standard XMLA. Cela permet à la fois de se connecter avec n'importe quel client utilisant le standard XMLA comme Pivot4J Analytics, Excel, JPivot (basé sur l'API du même nom), etc. On peut également changer de moteur OLAP tout en gardant cette infrastructure tant que celui-ci respecte le standard XMLA (Jedox PALO ou encore Microsoft SQL Server Analysis Services). De plus, le respect des standards de la Business Intelligence dans cette infrastructure permet une évolutivité plus simple.

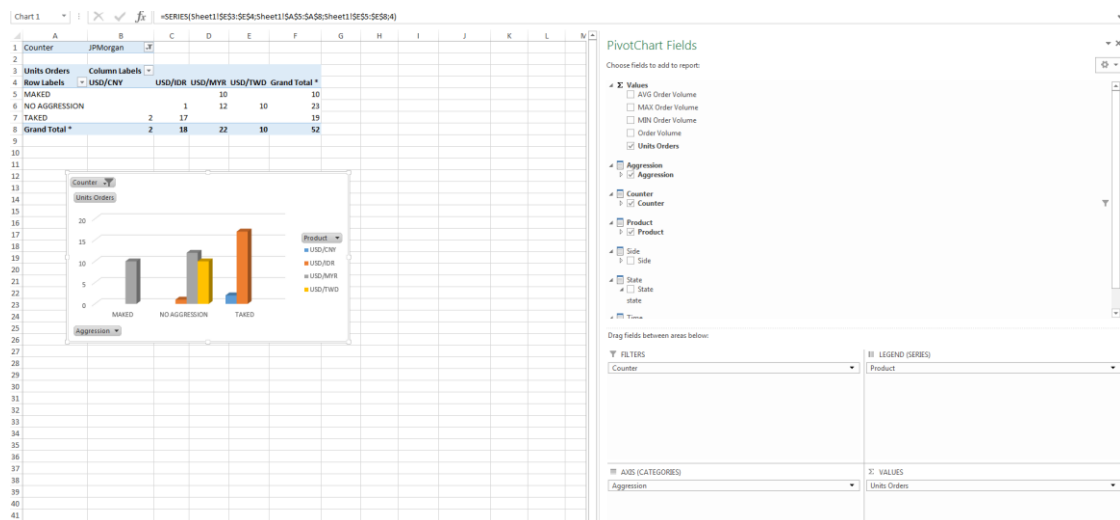


Figure 12: Connexion XMLA avec Excel

## Inconvénients de la solution proposée

La solution proposée nécessite la mise en place d'un entrepôt de données. Ce dernier est très coûteux en espace disque. De plus, la mise en place d'un moteur OLAP de type ROLAP a pour conséquence des temps de réponse assez longs sur certaines requêtes lorsque l'entrepôt de données sera très grand. Même si la gestion du cache de Mondrian permet diminuer le temps de réponse. Il faut, après avoir analysé les requêtes clientes les plus fréquentes, mettre en place des tables d'agrégations. Ces dernières vont être utilisées en lieu et place des tables de faits sur les requêtes les plus utilisées.

## Modèle en constellation

Le modèle de données « en étoile » est typique des structures multidimensionnelles stockant des données atomiques ou agrégées. Considéré comme un modèle dénormalisé, car il n'est pas en 3NF (3<sup>ème</sup> forme normale), le modèle en étoile permet une économie de jointures à l'interrogation, ce qui le rend optimisé pour les requêtes d'analyses.

La table située au centre de l'étoile est la table des faits ou mesures (ou encore métriques): ce sont les éléments mesurés dans l'analyse.

Les tables situées aux extrémités de l'étoile sont les tables de dimensions (ou encore axes d'analyse) ou niveaux de suivi. Ce sont les dimensions explorées dans l'analyse.

Le modèle en constellation est un modèle composé de plusieurs étoiles (ou flocon voir annexe : Modèle en flocon) dans lequel les tables de faits se partagent certaines tables de dimensions.

Ci-dessous le modèle conceptuel de donnée (MCD) du modèle en constellation proposé.

Ce modèle est implémenté par des tables SQL dans l'entrepôt de données. Ensuite, on effectue le « mapping » dans la description des données dans Mondrian (voir annexe: Mondrian: Déclaration et mapping d'un cube).

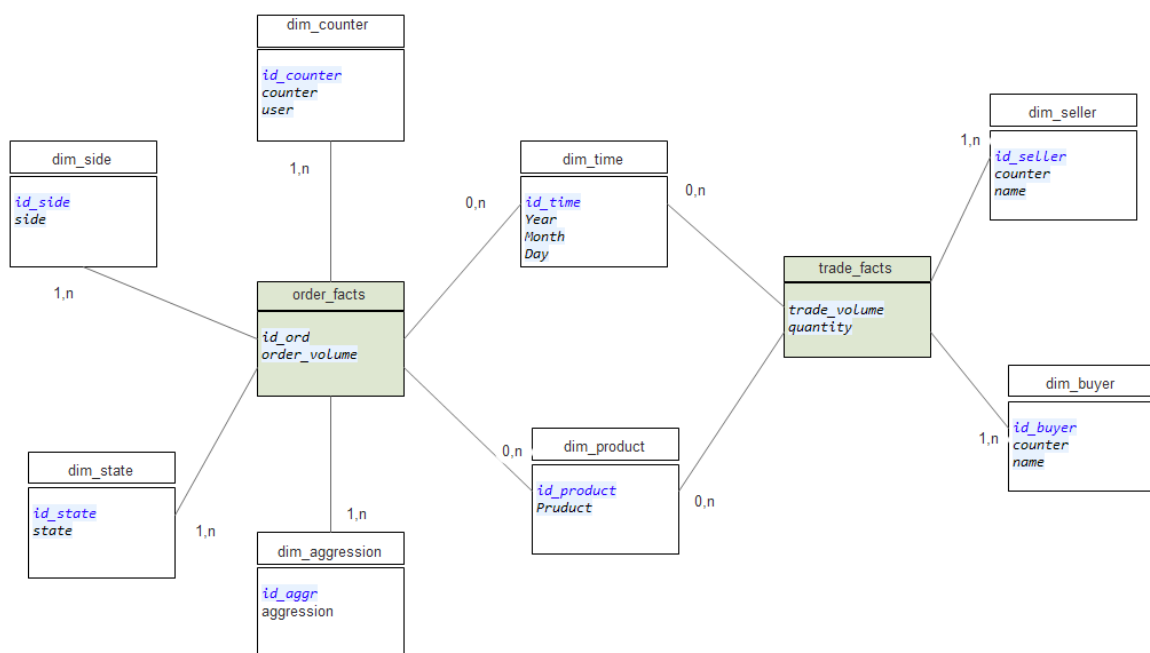


Figure 13: MCD du modèle en constellation

## Front end

Le front end est la partie visible d'une application pour un utilisateur lambda.

D'abord, l'interface front end proposée est composée de deux pages web différentes:

- « BICustom »
- « BIStats »

## Diagramme des cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet.

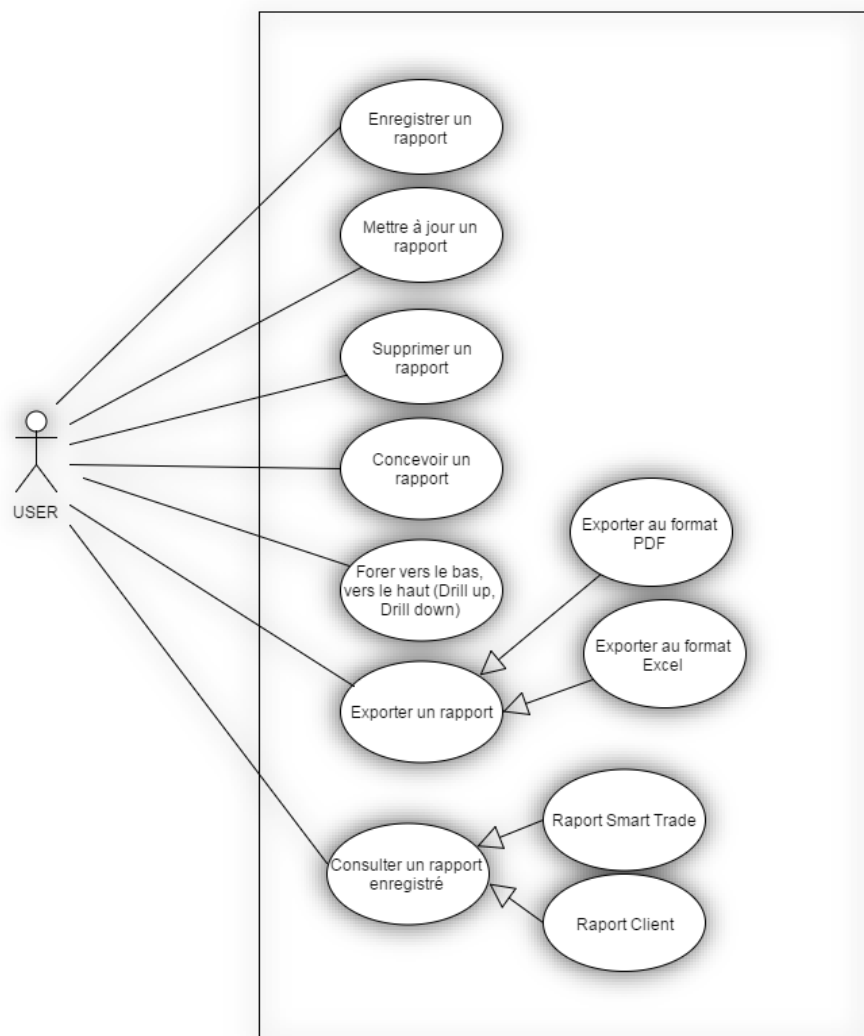


Figure 14: Diagramme des cas d'utilisations

Les deux pages ont en commun le fait d'exporter un rapport, le visualiser en diagramme ainsi que la navigation dans un cube. Les différents opérateurs de navigations dans un cube réalisés sont:

- Roll-up: Passage de mesures détaillées à résumées en remontant dans la hiérarchie de la dimension.
- Drill-down: Descendre dans la hiérarchie de la dimension.
- Drill Position: permet à l'utilisateur de percer uniquement la position d'un élément choisi.
- Drill Member: permet à l'utilisateur de percer la position d'un élément ainsi que tous les autres éléments de mêmes niveaux.
- Drill Replace: Permet de remplacer un élément par ses enfants.
- Rotate: Rotation des axes du cube pour fournir une vue alternative des données.
- Slicing: Extraction d'une tranche d'informations: Sélection d'une dimension pour passer à un sous-cube.
- Dice: Extraction d'un bloc de données: Sélection de deux ou plusieurs dimensions.

L'export des données se fait au format Excel ou PDF. Pour l'export en PDF, l'utilisateur a le choix sur le format du papier (A4, A3, etc.) ainsi que sur l'orientation du rendu (portrait, paysage, etc.).

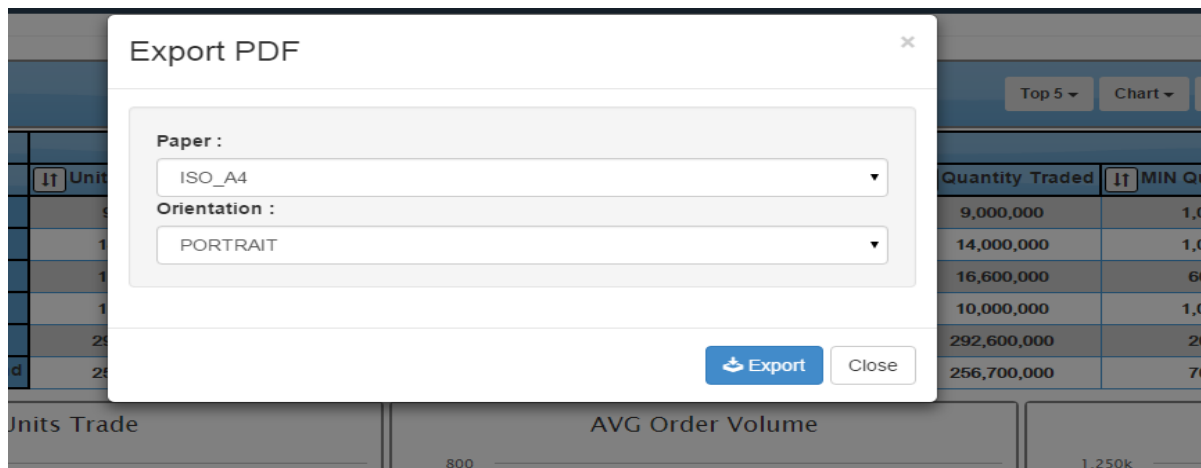


Figure 15: Export PDF

De plus, l'utilisateur pourra générer des diagrammes (en barre, en colonne, ou en secteur). Le rendu des diagrammes s'effectue à l'aide de l'API Java Script Highcharts.

## BICustom

La page « BICustom » permet de concevoir et d'enregistrer un rapport.

La conception d'un rapport est un des cas d'utilisation de l'application front end. Il se fait de manière ergonomique. En effet, l'utilisateur choisit une vue (représentation en cube). Ensuite il pourra glisser-déposer les dimensions et mesures sur les différents axes (colonne, ligne et filtre). Par la suite, il pourra soit enregistrer ou exporter le rapport qu'il a généré. En bas de la page s'affiche la requête MDX générée.

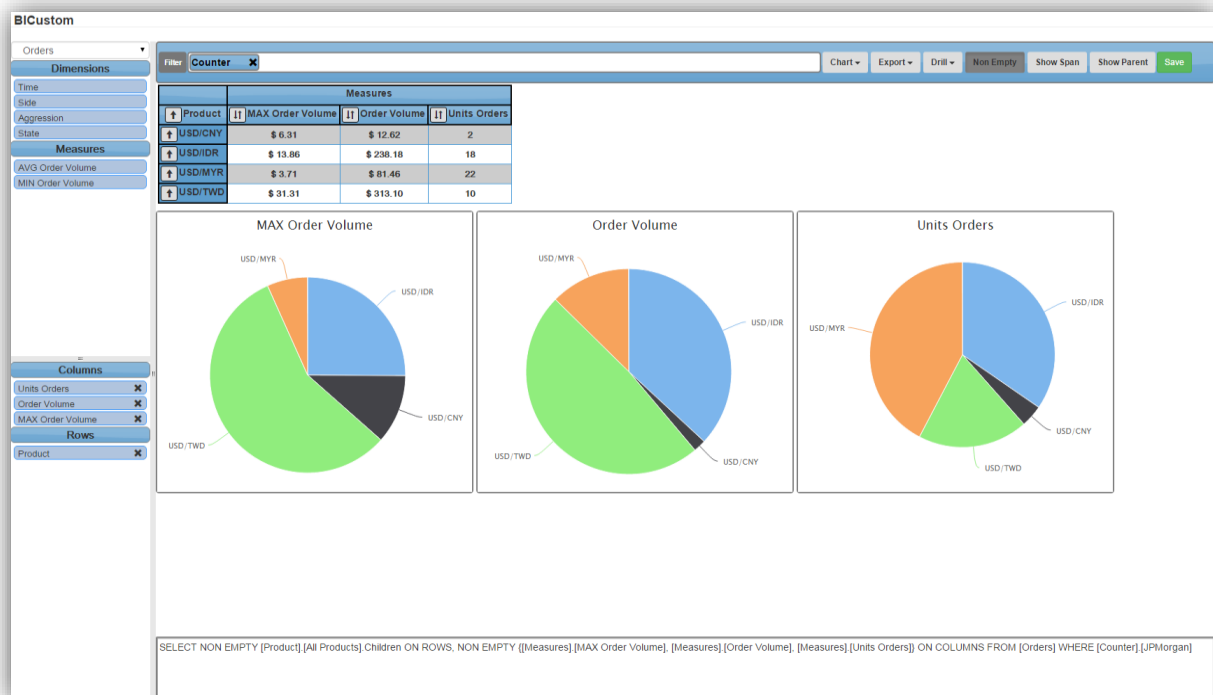


Figure 16: Page BICustom



## BISStats

La page « BISStats » permet de mettre à jour, de supprimer et de consulter un rapport enregistré. Les rapports enregistrés sont de deux sortes :

- les rapports clients: Ils sont générés par les clients avec la page « BICustom ».
- les rapports Smart Trade: Ils sont insérés (la requête MDX) en base par Smart Trade.

Il existe également dans cette page, une rubrique des meilleurs (les 5 meilleurs) vendeurs et acheteurs suivant la quantité, le volume et le nombre de trades.

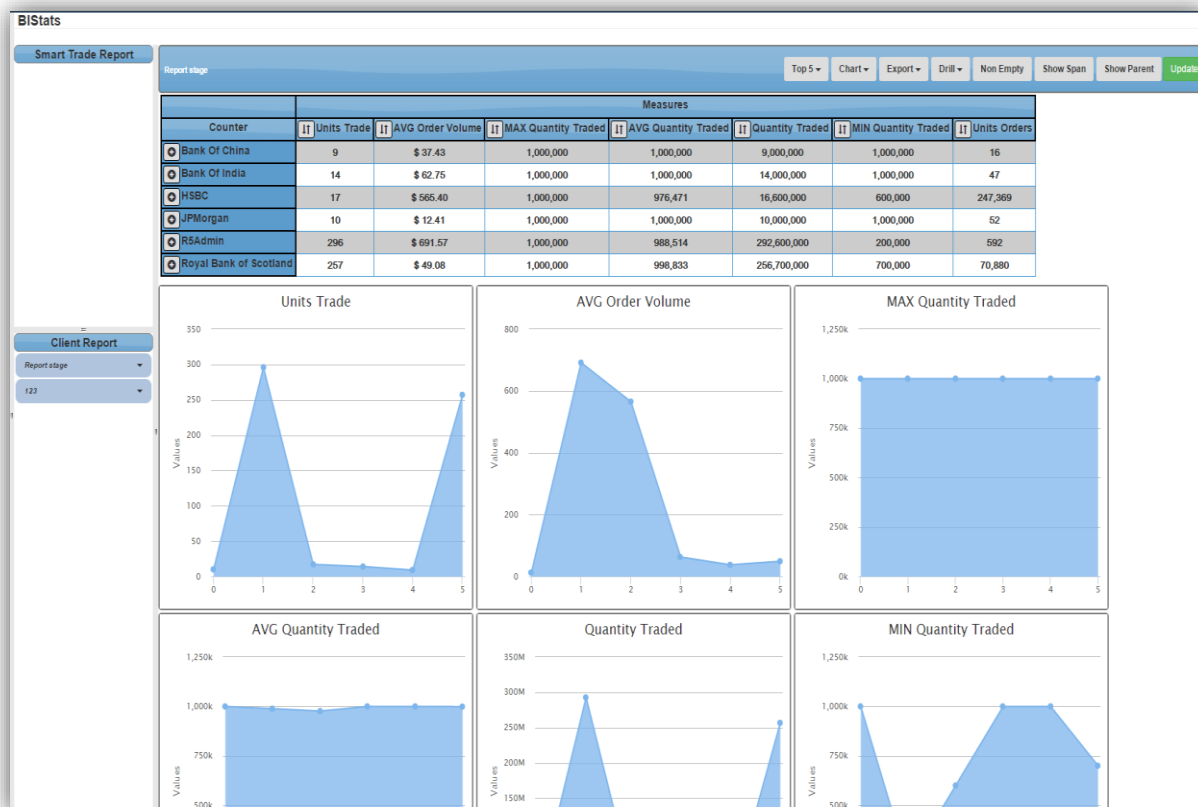


Figure 17: Page BISStats

# Logiciels, outils informatiques, protocoles et standards utilisés

## Java

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.



La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

## Highcharts

Permet de créer des diagrammes interactifs facilement pour des projets web.

Utilisé par des dizaines de milliers de développeurs et 61 des 100 plus grandes entreprises du monde, Highcharts est le plus simple API de diagrammes encore plus flexible sur le marché. Smart Trade dispose d'une licence Highcharts.



## JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs. Il a été créé en 1995. Il peut être directement intégré au sein des pages web, pour y être exécuté sur le poste client. C'est alors le navigateur Web qui prend en charge l'exécution de ces programmes appelés scripts.



## PonySDK

C'est un Framework open source cote front end qui englobe GWT et qui permet de créer des applications web rapidement en utilisant le langage java, développé par: Nicolas Ciaravola, Mathieu Barbier et Luciano Broussal.

## Spring

Spring est un socle pour le développement d'applications, principalement d'entreprises mais pas obligatoirement. Il fournit de nombreuses fonctionnalités parfois redondantes ou qui peuvent être configurées ou utilisées de plusieurs manières: ceci laisse le choix au développeur d'utiliser la solution qui lui convient le mieux et/ou qui répond aux besoins.



Spring est ainsi un des frameworks les plus répandus dans le monde Java: sa popularité a grandi au profit de la complexité de Java EE notamment pour ses versions antérieures à la version 5 mais aussi grâce à la qualité et la richesse des fonctionnalités qu'il propose:

- son cœur reposant sur un conteneur de type IoC (Inversion of Control) assure la gestion du cycle de vies des beans et l'injection des dépendances
- l'utilisation de l'AOP (programmation orienté aspect)
- des projets pour faciliter l'intégration avec de nombreux projets open source ou API de Java EE

Spring était un framework applicatif à ses débuts mais maintenant c'est une véritable plateforme composée du framework Spring, de projets qui couvrent de nombreux besoins et de middlewares.

Spring permet une grande flexibilité dans les fonctionnalités et les projets utilisés dans une application. Il est par exemple possible d'utiliser le conteneur Spring pour gérer de façon basique les beans sans utiliser l'AOP. Par contre, certains projets et certaines fonctionnalités ont des dépendances avec d'autres projets.

Spring est associé à la notion de conteneur léger (lightweight container) par opposition aux conteneurs lourds que sont les serveurs d'applications Java EE.

## Git

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2.



## Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.



Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, testing, gestion de configuration, reporting... Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.

Bien qu'Eclipse ait d'abord été conçu uniquement pour produire des environnements de développement, les utilisateurs et contributeurs se sont rapidement mis à réutiliser ses briques logicielles pour des applications clientes classiques. Cela a conduit à une extension du périmètre initial d'Eclipse à toute production de logiciel: c'est l'apparition du framework Eclipse RCP en 2004.

Figurant parmi les grandes réussites de l'Open source, Eclipse est devenu un standard du marché des logiciels de développement, intégré par de grands éditeurs logiciels et sociétés de services. Les logiciels commerciaux Lotus Notes 8, IBM Lotus Symphony ou WebSphere Studio Application Developer sont notamment basés sur Eclipse.

## CVS

CVS (sigle de Concurrent Versions System) est un système de gestion de versions créé en 1990, qui a été largement utilisé par les projets de logiciels libres.



Puisqu'il aide les sources à converger vers la même destination, on dira que CVS fait la gestion concurrente de versions ou de la gestion de versions concurrentes. Il peut aussi bien fonctionner en mode ligne de commande qu'à travers une interface graphique. Il se compose de modules clients et d'un ou plusieurs modules serveur pour les zones d'échanges.

Le premier logiciel qui a offert le service CVS s'appelle CVS, parmi les autres logiciels qui offrent un tel service il y a Cervisia, linCVS, TortoiseCVS, WinCVS, CVSNT.

## Gradle

Gradle est un moteur de production fonctionnant sur la plateforme Java. Il permet de construire des projets en Java, Scala, Groovy voire C++.



Gradle allie les atouts d'Apache Maven et d'Apache Ant: il allie l'utilisation de conventions à la manière de Maven (convention plutôt que configuration) avec la flexibilité d'Ant pour décrire les tâches de constructions, avec une cohérence forte dans l'interface de programmation des tâches.

## Mondrian

Mondrian est un moteur OLAP (Online Analytical Processing) écrit en Java par Julian Hyde qui permet la conception, la publication et le requêtage de cubes multidimensionnels. Mondrian permet l'exécution de requêtes en langage MDX sur des entrepôts de données s'appuyant sur des SGBDR (Système de gestion de base de donnée relationnel), d'où sa caractérisation de « ROLAP » (Relational OLAP). En matière de ROLAP, Mondrian est la référence open source.



## Olap4J

Olap4j est une API open source permettant d'accéder aux données OLAP. Il est une extension de JDBC. En effet Olap4j se veut un driver pour les serveurs ROLAP. Le projet Olap4j comporte également un driver pour XMLA permettant de se connecter à un serveur OLAP distant via XMLA.



## Pivot4J

Pivot4J fournit une API Java pour serveurs OLAP qui peuvent être utilisés pour construire un frontend BI. Il vise à tirer parti mature, mais maintenant discontinué de la base de code du projet JPivot pour en faire une API front end d'usage général qui est indépendant de toute implémentation graphique particulière contrairement à JPivot qui dépend des pages JSP.



Pivot4J fournit également une application entièrement fonctionnelle d'analyse OLAP construite au-dessus de sa bibliothèque centrale, avec un plugin pour la plateforme Pentaho BI qui peut être installé directement via le Market Place.

## Jira

JIRA est un système de suivi de bugs, un système de gestion des incidents, et un système de gestion de projets développé par Atlassian Software Systems.



## MDX

Le MDX (de l'anglais Multidimensional Expressions, « expressions multidimensionnelles ») est un langage de requête pour les bases de données OLAP, analogue au rôle de SQL pour les bases de données relationnelles. C'est aussi un langage de calcul avec une syntaxe similaire à celle des tableurs.

Le langage des expressions multidimensionnelles possède une syntaxe appropriée à l'interrogation et manipulation des données multidimensionnelles mémorisées dans un cube OLAP. Bien qu'il soit possible de traduire certaines expressions dans le langage SQL traditionnel, cela nécessite une syntaxe SQL souvent maladroite même pour des expressions MDX très simples. MDX a été adopté par une large majorité de fournisseur de la technologie OLAP et est devenu un standard de facto pour les systèmes OLAP.

## XMLA

XML for Analysis (XMLA en abrégé) est une norme d'accès aux données dans les systèmes analytiques, tels qu'OLAP ou le Data Mining. XMLA est basée sur d'autres normes de l'industrie telles que XML, SOAP et HTTP.

XMLA se compose de seulement deux méthodes SOAP:

- « Execute »: possède deux paramètres *command* (la commande à exécuter au format MDX, SQL ou DMX pour le Data Mining) et *Properties* la liste des propriétés de la commande
- « Discover »: permet à l'utilisateur de spécifier les données à découvrir avec les propriétés et restrictions possibles.

## Kettle

Pentaho Data Integration (PDI, également appelé Kettle) est la composante de Pentaho responsable du processus de l'extraction, Transformation et Chargement (ETL). Bien que les ETL soient souvent utilisés dans des environnements de DataWarehouse, PDI peut également être utilisé à d'autres fins:



- Migration des données entre les applications ou bases de données
- Exportation des données à partir des bases de données à des fichiers plats
- Chargement des données massivement dans des bases
- Nettoyage des données
- Intégration d'applications

Kettle est facile à utiliser. Chaque processus est créé avec un outil graphique (Spoon) et permet de spécifier ce qu'il faut faire sans écrire de code pour indiquer comment le faire.

PDI peut être utilisé comme une application autonome, ou il peut être utilisé comme une étape de la suite Pentaho. Comme ETL, PDI est le plus populaire dans le monde open source. PDI soutient un vaste éventail de formats d'entrée et de sortie, y compris les fichiers de texte, feuilles de données, et les moteurs de bases de données commerciales et libres. En outre, les capacités de transformation de PDI permettent de manipuler des données avec très peu de limitations.

## Bilan du travail

### Problèmes rencontrés

Tout d'abord, je tiens à dire que le stage s'est très bien déroulé. Je tiens à réitérer mes remerciements à toutes les personnes travaillant à Smart Trade Technologies pour l'accueil, l'ambiance et les expériences que j'ai acquises durant ce stage.

Un stage est avant tout une période d'apprentissage. Une période où l'on est confronté au métier. Passer des cours théoriques de la faculté à la pratique au sein de l'entreprise n'est pas facile. De ce fait, il y a eu une période de transition, dans laquelle il fallait faire une mise au point de ce qu'on savait déjà, de par les enseignements, mais surtout de ce qui était à découvrir.

Le sujet que l'on m'a proposé concerne un domaine auquel je ne suis pas familier. Pour le réaliser, j'ai dû utiliser certaines technologies comme l'API Pivo4J qui n'est pas trop documenté. En effet, j'ai dû consulter le code de cette dernière pour réellement comprendre son fonctionnement.

L'entreprise travaille dans un domaine bien précis. De ce fait, se familiariser à tous les termes techniques n'est pas facile. Cependant, au fil du temps, on arrive à s'y retrouver facilement et même à pouvoir les utiliser.

Plusieurs technologies internes à l'entreprise que je n'ai pas totalement maîtrisées ont été utilisées. De ce fait durant une partie du stage, il était important de consacrer du temps sur ces dernières. Certes, cela m'a fait perdre un peu de temps, et ralentit au niveau de la productivité. Mais au final, le travail fut achevé.

Et enfin, Smart Trade est une entreprise qui existe depuis plus de 10 ans. Plusieurs outils ont donc déjà été développés en interne. Au début, il était très difficile de se retrouver dans tout le code Smart Trade qui contient des dizaines de milliers de classes.



## Bilan du travail en entreprise

Le stage fut pour moi la deuxième expérience dans le monde de l'informatique. J'ai découvert dans ce stage comment s'organisent véritablement les informaticiens travaillant sur un même projet ou dans des projets indépendants.

J'ai également appris comment le fait d'être autodidacte était primordial dans le domaine de l'informatique. En effet, j'ai dû me documenter tout seul sur le monde de la « Business Intelligence », des technologies et des outils que je n'avais jamais utilisés auparavant. Tout cela dans un laps de temps assez court. De plus, j'ai été amené à découper un gros projet en petit module pour rendre le problème de départ plus facile à résoudre.

J'ai également été amené à réaliser des rapports, ainsi que des présentations sur mon travail pour faire part, à certains membres de l'entreprise, des fonctionnalités ou des solutions apportées. J'ai donc pu mesurer la difficulté de réaliser des rapports et des présentations simplifiés pour les personnes qui ne sont pas forcément dans la même branche d'activité, sur des travaux pourtant parfois techniques notamment pour parler des problèmes rencontrés.

Puis, j'ai acquis une expérience concernant le travail en entreprise en voyant comment tous les membres de l'entreprise travaillent ensemble bien que leur travail soit différent. J'ai également pu voir les réactions à avoir au moment des deadlines et les comportements de chacun à ces moments. Enfin, j'ai également acquis une expérience technique considérable. Le fait d'être confronté à des problèmes m'a amené à élargir mon champ de réflexion et ainsi élargir mon bagage connaissances.

## Bilan du travail personnel

A la fin de mon stage, je pense que le travail que j'ai fourni a plu aux membres de l'entreprise.

D'un point de vue informatique, ce stage m'a permis de découvrir le monde de la Business Intelligence, d'appréhender de nouveaux outils et technologies que j'ignorais auparavant.

L'ambiance conviviale dans l'entreprise Smart Trade et le sérieux qu'ont ses employés me conforte dans l'idée de travailler plus tard dans le domaine de l'informatique.

## Conclusion

Ce stage fut extrêmement bénéfique et formateur pour moi. J'ai pu avoir une approche professionnelle de l'informatique et comprendre l'importance des nombreuses compétences acquises pendant ces deux dernières années. Il m'a permis de me confronter à des problèmes techniques et de les résoudre. De plus, j'ai pu acquérir de nouvelles connaissances informatiques notamment en ce qui concerne le domaine de la Business Intelligence.

Grâce à la formation reçue durant ces deux dernières années, j'ai pu rapidement m'adapter dans l'entreprise, à des techniques jusque-là inconnues pour moi.

J'ai pu mener à bien un projet s'inscrivant dans l'ère du temps. Cela m'a permis de travailler en autodidacte tout en puisant dans les connaissances acquises durant toute ma période de formation en informatique et de les adapter à mon projet de stage.

## Lexique

**Instrument:** C'est un produit sur lequel s'effectue une transaction. Exemple: actions, obligations, monnaie, etc.

**Trader:** C'est une personne, travaillant généralement pour une institution financière, spécialisé dans l'achat et la vente d'instrument.

**Ordre:** Un ordre est une requête envoyée par un trader sur le marché dans laquelle il spécifie le prix, la quantité, le coté (achat ou vente) et l'instrument entre autre.

**Trade:** C'est lorsque deux ordres opposés (achat et vente) se correspondent. A partir de cela, la transaction s'effectue entre les deux émetteurs de ces ordres.

**Business intelligence:** L'informatique décisionnelle (en anglais business intelligence ou BI) est l'informatique à l'usage des décideurs et des dirigeants d'entreprises. Elle désigne les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données, matérielles ou immatérielles, d'une entreprise en vue d'offrir une aide à la décision et de permettre à un décideur d'avoir une vue d'ensemble de l'activité traitée.

**API:** En informatique, une interface de programmation (souvent désignée par le terme API pour Application Programming Interface) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur.

**GWT:** GWT ou Google Web Toolkit est un ensemble d'outils logiciels développé par Google, permettant de créer et maintenir des applications web dynamiques mettant en œuvre JavaScript, en utilisant le langage et les outils Java.

**JIRA:** est un système de suivi de bugs, un système de gestion des incidents, et un système de gestion de projets.

**SmartCC:** SmartCC est un outil développé avec PonySDK. Il est utilisé par les administrateurs.

**AJAX:** L'architecture informatique Ajax (acronyme d'Asynchronous JavaScript and XML) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.

## Annexes

### **MOLAP, ROLAP et HOLAP - Bases de données multidimensionnelles**

Les types de bases de données multidimensionnelles ou OLAP permettent des architectures multidimensionnelles et des approches différentes du décisionnel.

#### **MOLAP: Multidimensional OnLine Analytical Processing**

Le MOLAP, c'est de l'OLAP multidimensionnel. On stocke les données dans un cube qui est en fait une base de données multidimensionnelle. De cette façon, le concept de relationnel n'est plus présent. Pré calculer tous les croisements envisageables est l'objectif de cette base de données multidimensionnelle, de cette manière la restitution des données se fait de façon instantanée. Les données étant stockées, le temps gagné pendant la restitution des données sera considérable.

Les inconvénients de la méthode MOLAP: Le coût est important, en effet, elle nécessite souvent des licences pour les bases multidimensionnelles et des coûts pour le développement des cubes.

L'avantage de la méthode MOLAP: Le temps de réponse est extrêmement court, car la totalité des données est stockée au sein d'un cube.

Remarque: Les bases de données multidimensionnelles possèdent leur propre langage permettant de faire des requêtes, appelé le MDX, à opposer au SQL utilisé pour la base de données relationnelle.

#### **ROLAP: Relational Online Analytical Processing**

Le ROLAP, c'est de l'OLAP relationnel. L'obtention des données se fait via des tables relationnelles et des jointures qui vont avec celles-ci. Donc, la requête créée sera relativement complexe, selon la granularité, et sera d'une longueur plus ou moins importante. Comme le résultat n'est pas stocké, à chaque consultation, la requête devra être relancée.

Les différents inconvénients de la méthode ROLAP: Le temps de réponse est d'une longueur assez conséquente étant donné que les requêtes fonctionnent via des tables. Les bases sont donc utilisées à chaque relance du rapport.

Les avantages de la méthode ROLAP: Le coût est relativement faible, en effet, cette méthode utilise des ressources déjà existantes comme des ressources matérielles, des licences, etc.

## **HOLAP: Hybrid Online Analytical Processing**

L'HOLAP est un mélange du ROLAP et du MOLAP. On se sert du MOLAP lorsque l'on veut accéder pour aux données agrégées. Si l'on souhaite arriver à un niveau de détail plus important, nous utilisons le ROLAP. Un Exemple de restitution des données: Les données sont stockées dans un CUBE puis on fait la restitution via un outil de « reporting » comme BO par exemple. L'utilisateur pourra donc avoir accès à un rapport contenant les données issues du CUBE ainsi qu'à un autre rapport détaillé contenant les données en provenance des tables.

Les inconvénients de la méthode HOLAP: Elle est inutilisable en cas de complexité trop élevée des rapports ou qu'ils fassent appel à trop de croisements de données.

Les avantages de la méthode HOLAP: Un investissement financier moindre que la méthode MOLAP, en effet la partie développement sera beaucoup moins importante. De plus, le temps de réponse est relativement court.

## Modèle en flocon

Le modèle de données dit « en flocon » est une variante du modèle en étoile: chaque table de dimension est ré-normalisée pour faire apparaître la hiérarchie sous-jacente. La normalisation n'est pas indispensable car ni mises à jour ni suppressions ne sont effectuées directement sur l'entrepôt de données. L'intérêt principal du modèle en flocon réside dans le gain en espace de stockage qui est de l'ordre de 5 à 10 %.

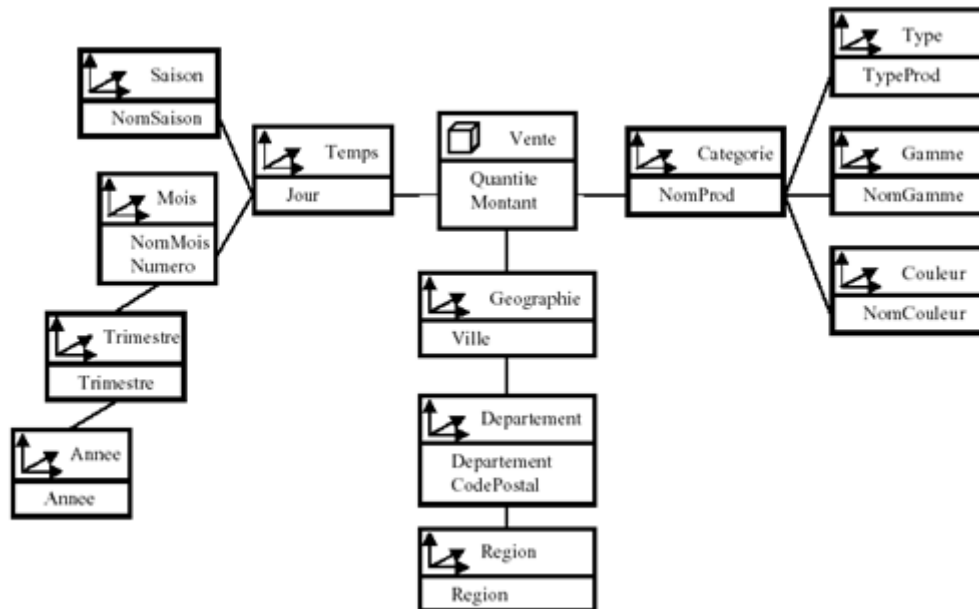


Figure 18: Exemple Modèle en flocon

# Mondrian: Déclaration et mapping d'un cube

```
<!-- Orders Cube -->
<Cube name="Orders">
  <Table name="order_facts" />
  <DimensionUsage name="Product" source="Product"
    foreignKey="id_product" />
  <DimensionUsage name="Time" source="Time"
    foreignKey="id_time" />
  <Dimension name="Counter" foreignKey="id_counter">
    <Hierarchy hasAll="true"
      allMemberName="All Counters"
      primaryKey="counter_id">
      <Table name="dim_counter" />
      <Level name="counter" column="Counter"
        uniqueMembers="true" />
      <Level name="user" column="User"
        uniqueMembers="true" />
    </Hierarchy>
  </Dimension>
  <Dimension name="Side" foreignKey="id_junk">
    <Hierarchy hasAll="true"
      allMemberName="All Sides"
      primaryKey="id_junk">
      <Table name="junk_dim" />
      <Level name="side" column="side"
        uniqueMembers="true" />
    </Hierarchy>
  </Dimension>
  <Dimension name="Aggression" foreignKey="id_junk">
    <Hierarchy hasAll="true"
      allMemberName="All Aggressions"
      primaryKey="id_junk">
      <Table name="junk_dim" />
      <Level name="aggression" column="aggression"
        uniqueMembers="true" />
    </Hierarchy>
  </Dimension>
  <Dimension name="State" foreignKey="id_junk">
    <Hierarchy hasAll="true" allMemberName="All states"
      primaryKey="id_junk">
      <Table name="junk_dim" />
      <Level name="state" column="state"
        uniqueMembers="true" />
    </Hierarchy>
  </Dimension>
  <Measure name="Units Orders" column="id_ord"
    aggregator="count" formatString="Standard" />

  <Measure name="Order Volume" column="order_volume"
    aggregator="sum"
    formatString="$ ###,###,###,##0.00" />
  <Measure name="MIN Order Volume" column="order_volume"
    aggregator="min"
    formatString="$ ###,###,###,##0.00" />
  <Measure name="MAX Order Volume" column="order_volume"
    aggregator="max"
    formatString="$ ###,###,###,##0.00" />
  <Measure name="AVG Order Volume" column="order_volume"
    aggregator="avg"
    formatString="$ ###,###,###,##0.00" />
</Cube>
```



## Table des figures

Figure 1: Organigramme Smart Trade.....	6
Figure 2: Liste des projets Smart Trade.....	8
Figure 3: Outils et service interne de Smart Trade.....	9
Figure 4: Exemple d'outils de reporting existant .....	11
Figure 5: Structure de l'API Olap4J .....	12
Figure 6: Metadata Olap4J.....	13
Figure 7: Tableau croisé dynamique (avec Pivot4J et PonySDK) .....	14
Figure 8: Infrastructure Pentaho BI Server .....	16
Figure 9: Infrastructure proposée .....	17
Figure 10: ETL (Insertion dans le cube des ordres) .....	17
Figure 11: Diagramme de déploiement .....	18
Figure 12: Connexion XMLA avec Excel .....	19
Figure 13: MCD du modèle en constellation .....	20
Figure 14: Diagramme des cas d'utilisations .....	21
Figure 15: Export PDF .....	22
Figure 16: Page BICustom.....	23
Figure 17: Page BIStats .....	24
Figure 18: Exemple Modèle en flocon .....	38

## Webographie

<http://www.pentaho.fr/>

<http://mondrian.pentaho.com/documentation/schema.php>

<http://community.pentaho.com/projects/data-integration/>

<http://www.pivot4j.org/>

<http://www.olap4j.org/>

<https://msdn.microsoft.com/en-us/library/ms145506.aspx>

<https://gradle.org/>